



A parallel two mesh method for speeding-up processes with localized deformations: application to cogging

Mohamad Ramadan, Lionel Fourment, Hugues Dignonnet

► To cite this version:

Mohamad Ramadan, Lionel Fourment, Hugues Dignonnet. A parallel two mesh method for speeding-up processes with localized deformations: application to cogging. 12th ESAFORM Conference on Material Forming, Apr 2009, Enschede, Netherlands. pp.Pages 581-584, 10.1007/s12289-009-0440-x . hal-00509412

HAL Id: hal-00509412

<https://hal-mines-paristech.archives-ouvertes.fr/hal-00509412>

Submitted on 12 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A PARALLEL TWO MESH METHOD FOR SPEEDING-UP PROCESSES WITH LOCALIZED DEFORMATIONS: APPLICATION TO COGGING

M.Ramadan*, L.Fourment, H.Digonnet

Mines ParisTech, CEMEF - Centre For Material Forming, CNRS UMR 7635,
1 rue Claude Daunesse, 06904 Sophia Antipolis, France

ABSTRACT: In order to reduce the very long computational time required for the simulation of incremental processes such as cogging or ring rolling, a Bimesh method is proposed. It consists in using different finite element meshes for the resolution of the different physical problems: a main fine mesh to store the results and to carry out the thermal computations, and an intermediate coarser mesh for the mechanical calculations. It makes it possible to take advantage of the localised deformations that characterize incremental processes to coarsen the mechanical mesh and consequently reduce the computational time. After presenting the main components of these *Bimesh* method, the building of the embedded meshes and the data transfer between the meshes, its extension for parallel calculations are discussed, before analysing the obtained speed-ups for several cogging applications, both in sequential and in parallel.

KEYWORDS: Bimesh, Meshing, Coupled Problems, Parallel Calculations, Parallel Meshing, Forging, Cogging

1 INTRODUCTION

Incremental metal forming processes such as cogging and ring rolling (see figure 1) are characterized by cycles of loadings and unloadings that are successively applied to every region of the component with the same set of tools. These types of processes are used in a wide range of industrial applications. They have many advantages that make them attractive. Unlike many other forging processes, where every component requires manufacturing specific tools, they operate with relatively simple tools that can be used for a wide range of products. Another advantage of these processes, and especially for very large components, is the need of significantly reduced forces. Eventually, numerical simulation can be used to optimize the forming sequences in order to keep proper material formability and to obtain good final properties.

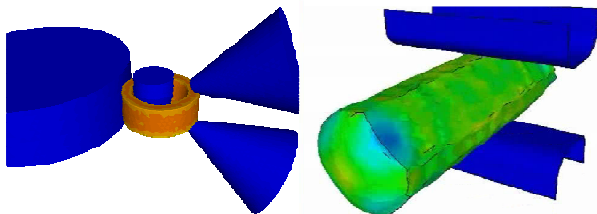


Figure 1: Simulations of incremental processes: ring rolling (left side) and cogging (right side).

Like other forming processes, incremental forming has significantly evolved during the last decades, especially

regarding its numerical simulation, which is now playing a central role in the designing of the manufacturing sequence. Despite of the process simplicity; in terms of tooling; its numerical simulation is made difficult by the very large number of strokes, which results into very large computational time (several days) even with parallel computers (up to nine, for instance). In order to speed-up the calculations and make such simulations feasible, several numerical methods have been investigated in the literature, such as highly efficient solvers based on the multigrid method. However, the obtained computational time reductions, ranging between 1.3 and 2, are not enough for such processes. It so appears necessary to take into account the process specificities for larger speed-ups.

The main characteristic of incremental processes is that at any instance in time, the deformation occurs within limited region of the workpiece. Based on this property Kim et al. [1] proposed to use two different meshes to simulate the process of ring rolling: an Actually rotating Mesh System (AMS) and a Spatially fixed Mesh System (SMS). The SMS is a coarse mesh that is only refined between the rolls; it is used to solve the thermal and mechanical equations. The storage of results is carried out on the AMS which is a uniformly fine mesh. This method is also used in [2], but it is improved by the so-called hybrid mesh model. It consists in keeping the nodes of the deformation zone of the AMS the same as the nodes of the SMS. Hirt et al. [3] used a similar method to simulate the cogging process; they call it *multimesh*.

* Corresponding author: Mohamad Ramadan, Mines ParisTech, CEMEF, CNRS UMR 7635, 1 rue Claude Daunesse, 06904 Sophia Antipolis, France, phone: +33 (0)4 93 67 89 31, fax: +33 (0)4 92 38 97 52, mohamad.ramadan@mines-paristech.fr

Although these methods allow reducing the computational time, they have three main drawbacks: 1) the storage mesh is not remeshed, which does not allow using the method with very large material deformations, 2) the variables that evolve all over the mesh, such as temperature, are only computed on the coarse mesh, which provides inaccurate results outside the deformation zone, 3) the authors have not presented how to derive them for parallel calculations.

In this paper, a so-called Bimesh method is proposed to reduce computational time of incremental processes. It can be regarded as an enhancement of previously presented methods in order to avoid the mentioned disadvantages.

2 THE BIMESH METHOD

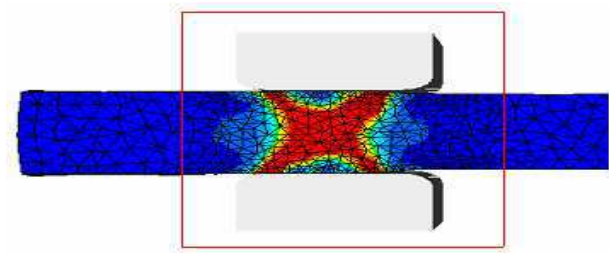


Figure 2: Isovalues of equivalent strain rate for an open die forging simulation with localised deformation.

2.1 PRINCIPLE

The Bimesh method is developed within the Cimlib library [4], which is a parallel finite elements code dedicated to material forming simulation and developed within the CIM group of Mines ParisTech, CEMEF.

As in [1-3], it is based on the particularity of processes that have localized deformations. For instance, Figure 2 shows the localized equivalent strain rate for a cogging simulation. Its main idea is to divide the calculations into two parts: the “Mechanical Calculation”, which is the most expensive, and the “Thermal Calculation”, which is highly less expensive, and which cost can almost be neglected in a first approach. The Mechanical Calculation is done on an intermediate coarse mesh, the Mechanical Mesh (MM), which is obtained by a special derefinement. This reduction of nodes number provides a computational time decrease. The Thermal Calculation is performed on a reference mesh, also called the Thermal Mesh (TM), which has a uniformly fine mesh size, as the thermal resolution requires the same fine size all over the component. The TM is also the main mesh, which is used to store all the thermo-mechanical data, so providing a very good accuracy for the history variables (temperature, strains, stresses, ...). The separation of the mechanical calculation from the thermal calculation is one of the main innovation of the Bimesh method with respect to previous methods [1-3]; it so allows keeping a high accuracy on the thermal calculation (which was considered as one of the main limitation of previous approaches [3]). For instance, Figure 3 shows that temperature uniformly evolves during cogging, unlike

the mechanical variables that are highly localized and which variations are negligible outside the deformation area (see Figure 2).

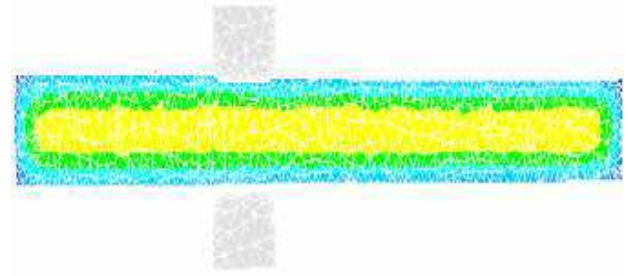


Figure 3: Temperature isovalues (variables evolving over all the mesh) during cogging.

The Bimesh method can also be regarded a two-grid solver dedicated to the mechanical calculation. The thermal and mechanical problems are expected to be solved on the TM, but as the TM is too fine in the undeformed areas, before carrying out the finite element discretization of the mechanical equations, the thermo-mechanical data of the TM are projected onto the MM; then the mechanical problem is solved on the MM, and all calculated data are extrapolated back on the TM. From this standpoint, the MM is just an intermediate grid that allows carrying out mechanical calculations that are as accurate, but on a better adapted grid with less degrees of freedom.

2.2 ADAPTATION OF THE MESH MM

The MM is derived from the TM by a particular mesh coarsening procedure that allows keeping the best accuracy. This coarsening is related to the tools displacements. It consists in dividing the domain into two zones, the Deformation Zone (DZ) and the Weak Deformation Zone (WDZ). The DZ represents the area of expected large deformations, which is included between the contact areas of the tools (see Figure 4). The DZ moves through the mesh along with the movement of the tool. The definition of the DZ is an important parameter of the Bimesh method, because it has to be large enough to capture the details of the deformation. Therefore, the number of nodes and hence the computational time significantly depends on the dimensions of this zone, which can easily be determined in advance for most forming process. In the present applications of open-die forging, the DZ has the same length as the tool with a margin of 20 %; it has the workpiece height and width. This choice results from previous numerical studies which have shown that the deformations outside this zone are negligible.

Nodes and elements inside the DZ are “frozen” during the mesh coarsening procedure; it means that they are identical in the MM and TM. The WDZ is the complementary part of the DZ; it contains all the nodes and the elements not belonging to DZ. In the MM, the WDZ is obtained by a node nested coarsening procedure; it consists in derefining the mesh by removing nodes of the TM without generating new nodes or moving their

positions. The key advantage of this derefinement procedure is that all the nodes of the MM belong to the TM, and that in the DZ all nodes and elements are exactly the same for both meshes. These properties of the MM increases the precision of the data transfer between MM and TM, as will be explained in the next section.

2.3 DATA TRANSFER

The simulation of an incremental process is generally based on a thermo-mechanical formulation, so it contains several mechanical parameters that depend on the thermal calculation and vice versa. Due to this, a transfer of data between the two meshes, MM and TM, is necessary to exchange the information between the meshes.

The thermo-dependent mechanical variables of the mechanical equations, such as the material consistency; are calculated as functions of the temperature (and of the cumulated strains, ...) which have to be transferred from the TM onto the MM. Nodal interpolated values like temperature are so exactly transferred, because all nodes of the MM belong to the TM. The other variables calculated and stored at the integration points (like the cumulated strains and the stresses) are also exactly transferred exactly within the DZ; because all the elements of this zone belong to the TM. In the WDZ, these variables need to be interpolated. However, it should be noticed that they are not expected to play a very important role in the mechanical calculation, because the deformation is almost negligible in this zone; So, the resulting interpolation errors are not expected to have noticeable influence on this calculation. Once the mechanical equations are solved, a new data transfer is carried out in the opposite sense, from the MM to the TM. Again, the transfer of nodal variables and of variables calculated at integration points is perfectly exact inside the DZ. Outside the DZ, in the WDZ, an interpolation procedure is necessary for both types of variables. However, it must be noticed that the quantity to interpolate (velocities, displacements, increments of strains and stresses) are expected to be very small, so that the resulting transfer error should be quite negligible at a global level.

3 RESULTS

3.1 RESULTS VALIDATION

From an industrial point of view, the two main parameters that make a finite element code reliable and efficient are its accuracy and its required computational time. Unfortunately these two objectives are antagonistic. Therefore, it is quite important to check the quality of the Bimesh method along with the resulting speed-up. To assess the quality of the results, comparisons are performed between the Bimesh method and reference calculations carried out using only the TM. The precision is more than satisfactory and the results look almost identical as can be seen in Figure 4, which shows the equivalent strain rates respectively obtained

by the Bimesh method (upper part) and the reference (mono-mesh) method (lower part).

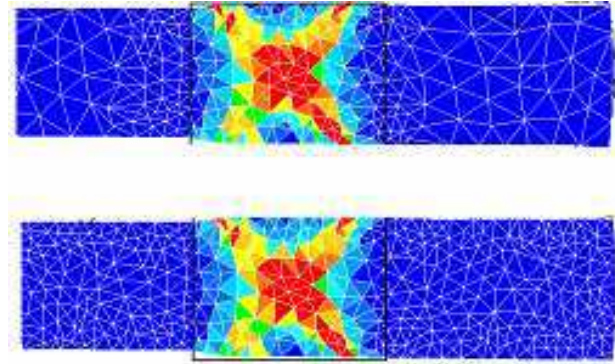


Figure 4: Isovalues of equivalent strain rate given by the Bimesh method (upper part) and by the standard method (lower part)

3.2 BIMESH SPEED-UP

The speed-up of the Bimesh method mainly depends on two parameters: 1) the dimensions of the DZ, 2) the mesh size in the zone WDZ. As a matter of fact, it so depends on the process features (ratio between the DZ and the WDZ). However, an estimation of the expected speed-up can be derived from the computational cost of the utilised iterative non linear solver, which cost is

$$O(N^{1.3}): \left(\frac{N_{TM}}{N_{MM}} \right)^{1.3}, \text{ where } N_{TM}, N_{MM} \text{ respectively are}$$

the number of degree of freedom of the TM, and the MM.

3.3 APPLICATIONS

We consider 10 increments of deformation (corresponding to 4% of material deformation for the utilized forging velocity and time steps) of the first operation of a multi-pass cogging simulation. In order to study the asymptotic cost of the Bimesh method, the simulations are carried out on 3 meshes having an increasing number of degrees of freedom. The obtained speed-up are shown in table 2: they are ranging between 3.8 and 6.

Table 1: Bimesh Speed-up (first pass of the simulation)

N_{TM}	N_{MM}	$\frac{N_{TM}}{N_{MM}}$	Speed up
12,600	3,500	3.6	3.8
23,500	7,300	3.2	4.2
53,500	16,500	3.2	5.7

Another study is carried out for the same 10 increments of the fourth operation. The results are shown in table 3: they are even better, with a speed-up reaching 7.5. This is due to the aspect ration of the fourth operation, which is more favourable; the DZ is smaller with respect to the entire mesh, and the ratio N_{TM}/N_{MM} is larger.

Table 2: Bimesh Speed-up (fourth pass of the simulation)

N_{TM}	N_{MM}	$\frac{N_{TM}}{N_{MM}}$	Speed up
9,500	2,400	3.9	5.9
24,000	5,600	4.2	6.2
54,500	12,400	4.4	7.5

4 PARALLEL ASPECTS

The CIMLIB is a fully parallelized library [4] that is based on a SPMD approach and mesh partitioning techniques. The domain mesh is so partitioned in as many domains as available processors, and each processor only contains a part of the global mesh.

4.1 PARALLEL REMESHING

The remeshing operations are performed using the MTC mesh generator [5], which fully works in parallel. The procedure is iterative. It starts by partitioning the mesh into sub-domains that are separated by interfaces. During the first remeshing iteration, which is done in parallel on each processor, the sub-domain interfaces are kept unchanged, which avoid any kind of communication between sub-domains. Then, a new partitioning is carried out to move the interfaces and to enable their remeshing during the next iteration. The algorithm iterates until all the elements have been remeshed.

The transfers between the TM and MM are also carried out in parallel. They can be quite complex because these two meshes are independently partitioned, so there is no reason for the DZ of the MM to belong to the same processor as the corresponding zone of the TM. Preliminary studies have shown that the transfer method requires to be slightly enhanced for the Bimesh method. Therefore, in the following results, the transfer times that have been used to estimate the Bimesh speed-ups have been approximated very accurately by an approximation of the forthcoming parallel transfer algorithm.

4.2 APPLICATION

Table 3: Parallel Bimesh Speed-up

	CPU [s]	Speed-up
1 Processor	654.8	1
2 Processors	411.6	1.6
4 processors	227.8	2.9
8 Processors	155	4.2

The parallel Bimesh method is applied to the same cogging simulation with a mesh of 23,000 nodes. Calculations are carried out on 1, 2, 4 and 8 processors. The results are shown in Table 3, the speed up on 2 processors is 1.6, which is an expected result because of

the drop of performance of the parallel preconditioner. The speed-up between 2 and 4 processors is equal to 1.8. With 8 processors the speed-up drops because the number of degrees of freedom in two small on each processor.

Table 4: Bimesh Speed-up in parallel

	Standard CPU [s]	Bimesh CPU [s]	Speed-up
1 Processor	2,774	655	4.2
2 Processors	1526	412	3.7
3 Processors	710	228	3.1
4 Processors	518	155	3.3

The parallel speed-up of the Bimesh method is evaluated by comparing the parallel computational time for both the Bimesh and the standard methods; the results are presented in table 4. It can be noticed that the obtained speed-up is still larger than 3, whatever the number of processors.

5 CONCLUSIONS

Based on the characteristics of processes with localized deformations, the Bimesh method allows decreasing the computational time by factors ranging between 4 to 7 in cogging, while maintaining the accuracy of both the thermal and mechanical calculations. In parallel, the speed-up seems to be slightly reduced, but still remains quite high.

REFERENCES

- [1] Kim N., Machida S., and Kobayashi S. Ring rolling process simulation by the three dimensional finite-element method. *International Journal of Machine Tools and Manufacture*, 30: 569-577, 1990
- [2] Hu Z. M., Pillinger I., Hartley P., McKenzie S., Spence P. J. Three dimensional finite-elements modelling of ring rolling. *Journal of Materials Processing Technology* 45: 143-148, 1994
- [3] Hirt G, Kopp R., Hofmann O., Franzke M., and Barton G. Implementing a high accuracy Multi-Mesh Method for incremental Bulk Metal Forming. *CIRP Annals-Manufacturing Technology*, 56: 313-316, 2007
- [4] Digonnet H., Coupez T., Object-oriented programming for fast and easy development of parallel applications. forming processes. In: *Second MIT conference on computation fluid and solid mechanics*, Pineridge press 1922-1924, 2003
- [5] Coupez.T., A mesh improvement method for 3D automatic remeshing, *Numerical grid generation in computational fluid dynamics and related fields*, 615-624, 1994